

Projet BE

TP de base HC-SR04

MASTER-EEA-SME

Année 2022-2023

Réalisé par :
Souleimane Erroussi

Sommaire:

Introduction.....	3
I. NUCLEO-476-RG.....	4
1. schéma fonctionnel.....	4
II. CAPTEUR ULTRASON (HC-SR04).....	5
A. Caractéristiques.....	5
B. Broches de connection.....	6
C. Spécifications et limites.....	6
D. Schéma électrique.....	6
E. Fonctionnement.....	7
F. Distance de l'objet.....	7
G. programme du capteur.....	8
H. visualisation picoscope.....	9
I. Validation Expérimentale.....	9
III. Conclusion.....	10

Introduction

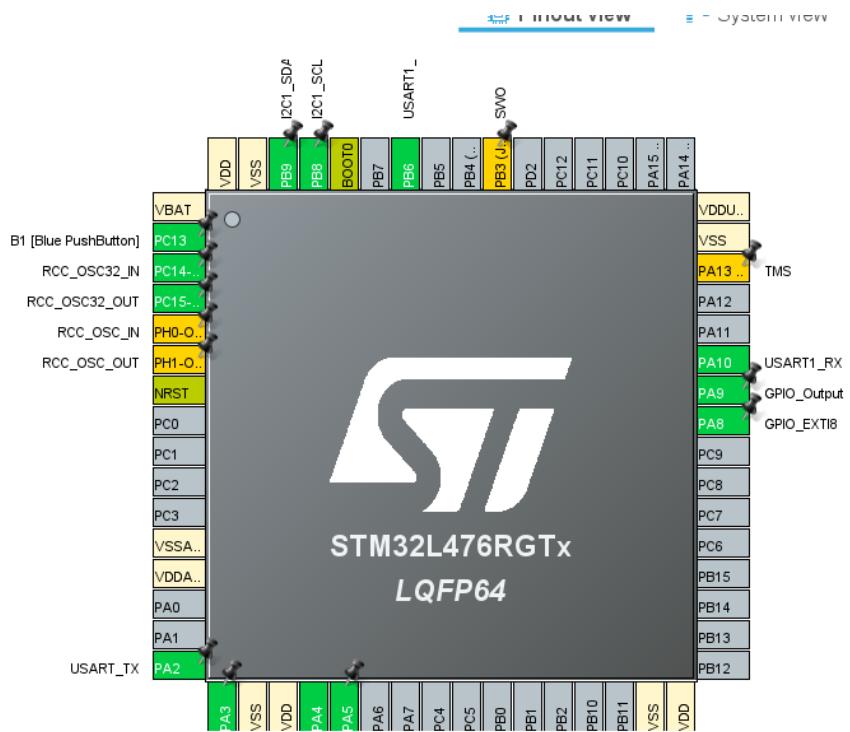
Dans le cadre de notre première année de master SME à l'université de paul sabatier, il nous est proposé un projet séparé en 2 partie à réaliser en 32h, un premier projet de base nous permettant de comprendre, nous familiariser avec un nouveau microcontrôleur(nucleo-476-rg) , nouveau logiciel(stm32cubeIDE) . et par la suite de réaliser un système qui correspond au projet de BE.

Notre projet BE est la Ruche connectée, permettant de connaître grâce au capteur de distance et et le capteur de température de connaître la température de la Ruche et permettant de surveiller la ruche si une personne s'approche de la ruche en envoyant ces informations dans un centre de surveillance grâce au capteur LoRa qui indique .

Pour réaliser ce bureau d'étude nous avons d'abord pris connaissance des capteurs (HC-SR04, DHT22) et de leur utilisation, du logiciel STM32cubeIDE et du capteur Grove LoRa.

Ce projet a été réalisé par souleimane Erroussi et Jidan Bilal

I. NUCLEO-476-RG



Pour commencer, nous devons configurer la communication avec l'écran LCD, qui utilise le protocole I2C à 4 fils. Sur le microcontrôleur, nous devons définir les broches SDA et SCL pour établir la communication avec l'écran LCD. Dans le schéma précédent, nous pouvons



PA9 :GPIO output (pin Trig)

PA8 :GPIO input (pin Echo)

PA9 est activé pour la broche du trigger du capteur ultrason permettant d'envoyer l'ultrason et la broche PA8 est activé pour recevoir l'information du capteur ultrason.

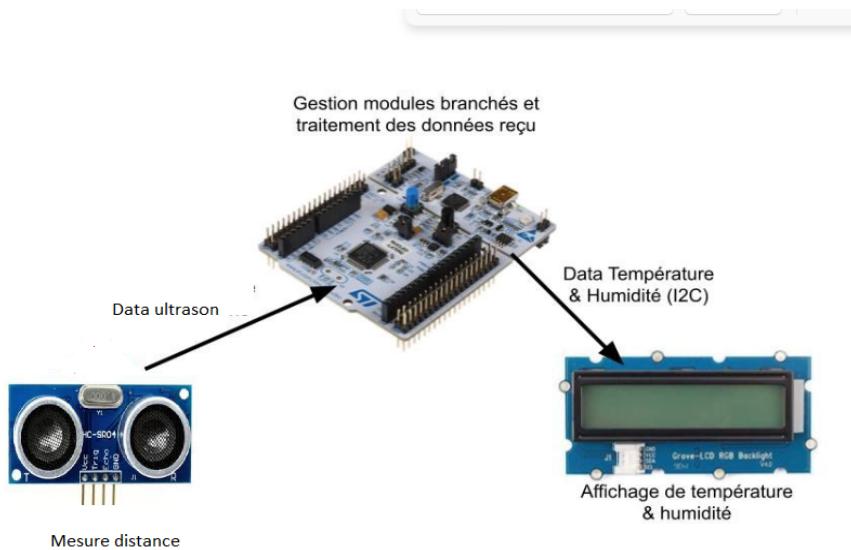
PA10: USART1 input(pin RX)

PB6: USART1 output(pin TX)

Nous avons activé la broche PA10 pour déclarer l'entrée Echo (Rx) et nous avons activé la broche PB6 pour déclarer la sortie du trigger (TX) du capteur LoRa sur l'USART1 car l'USART2 est utilisé pour la connection entre le pc et le microcontrôleur.

Enfin, il est important de configurer un timer en microsecondes, car cela est nécessaire pour assurer le bon fonctionnement du capteur HC-SR04. Ainsi, en prenant en compte ces différentes configurations, nous pourrons établir une communication correcte entre le microcontrôleur STM32, l'écran LCD et le capteur HC-SR04.

1. schéma fonctionnel

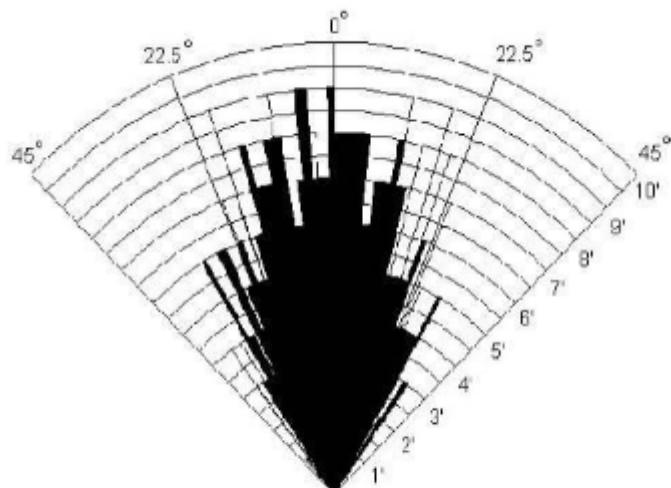


II. CAPTEUR ULTRASON (HC-SR04)

Le capteur HC-SR04 utilise les ultrasons pour déterminer la distance d'un objet. Il offre une excellente plage de détection sans contact, avec des mesures de haute précision et stables. Son fonctionnement n'est pas influencé par la lumière du soleil ou des matériaux sombres.

A. Caractéristiques

- Dimensions : 45 mm x 20 mm x 15 mm
- Plage de mesure : 2 cm à 400 cm
- Résolution de la mesure : 0.3 cm
- Angle de mesure efficace : 15 °
- Largeur d'impulsion sur l'entrée de déclenchement : 10 µs
- (Trigger Input Pulse width)



*Practical test of performance,
Best in 30 degree angle*

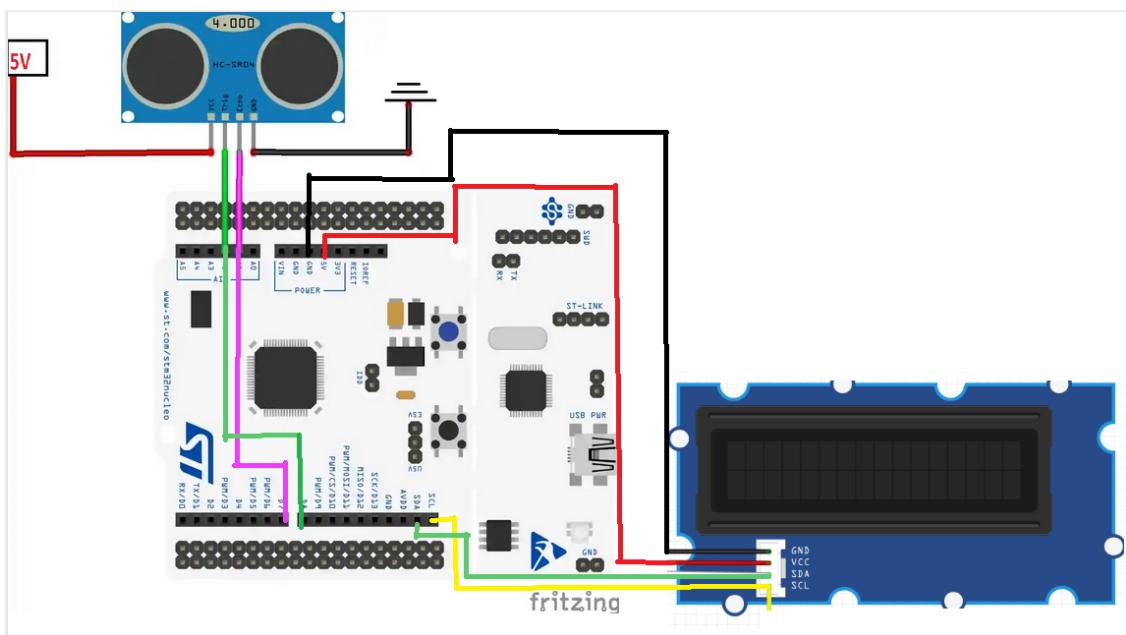
B. Schéma électrique

La broche Trig est branché sur la broche digital 8 du microcontrôleur

La broche Echo est branché sur la broche digital 7 du micro.

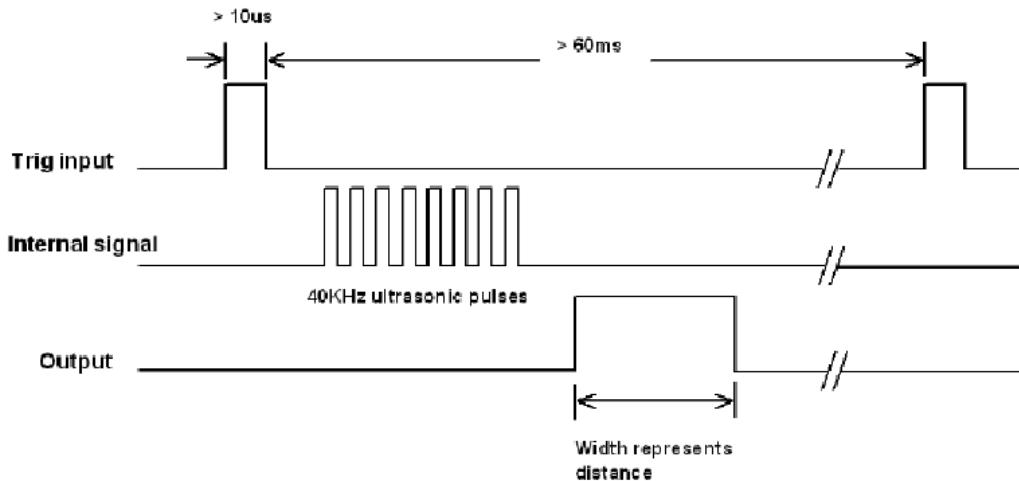
On a les broches de l'écran qui sont relié sur les broche du micro SDA avec SDA
SCL avec SCL....

Le cap



C. Fonctionnement

Pour déclencher une mesure, il faut présenter une impulsion "high"(5 V)d'au moins 10 μ s sur l'entrée "Trig".Le capteur émet alors une série de 8 impulsions ultrasoniques à 40 kHz,puis il attend le signal réfléchi.Lorsque celui-ci est détecté, il envoie un signal "high" sur la sortie "Echo", dont la durée est proportionnelle à la distance mesurée.



D. Distance de l'objet

La distance parcourue par un son se calcule en multipliant la vitesse du son, environ 340 m/s (ou 34'000 cm/1'000'000 µs) par le temps de propagation, soit : $d = v \cdot t$ (distance = vitesse · temps). Le HC-SR04 donne une durée d'impulsion en dizaines de µs. Il faut donc multiplier la valeur obtenue par 10 µs pour obtenir le temps t. On sait aussi que le son fait un aller-retour. La distance vaut donc la moitié.

$$d = 34'000 \text{ cm}/1'000'000 \mu\text{s} \cdot 10\mu\text{s} \cdot \text{valeur / 2}$$

en simplifiant

$$d = 170'000 / 1'000'000 \text{ cm} \cdot \text{valeur}$$

Finalement, $d = 17/100 \text{ cm} \cdot \text{valeur}$

La formule $d = \text{durée}/58 \text{ cm}$ figure aussi dans le manuel d'utilisation du HC-SR04 car la fraction 17/1000 est égale à 1/58.8235. Elle donne cependant des résultats moins précis.

Note : A grande distance, la surface de l'objet à détecter doit mesurer au moins 0.5 m²

E. programme du capteur

La fonction `HAL_GPIO_EXTI_Callback` est une fonction de rappel (callback) utilisée dans le contexte des interruptions de broches GPIO. Lorsqu'une interruption externe est déclenchée sur la broche GPIO spécifiée (`GPIO_Pin`), le microcontrôleur exécute automatiquement cette fonction.

Le but de cette fonction est de réagir à l'événement d'interruption sur la broche GPIO D8.

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    // Déclaration et initialisation des variables
    float distance = 0;
    char chaine[50];

    // Réinitialise le compteur htim2 à 0
    __HAL_TIM_SET_COUNTER(&htim2, 0);

    // Boucle tant que le GPIO pin PA8 est activé
    while(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_8) == GPIO_PIN_SET)
    {
        // Calcule la distance en utilisant la valeur du compteur htim2 divisé
        distance = __HAL_TIM_GET_COUNTER(&htim2) / 58;
    }

    // Formate la valeur de la distance dans la chaîne chaine
    sprintf(chaine, "distance=%2.f cm", distance);

    // Positionne l'affichage LCD à la première ligne et à la première colonne
    lcd_position(&hi2c1, 0, 0);

    // Affiche la chaîne chaine sur l'affichage LCD
    lcd_print(&hi2c1, chaine);
```

F. programme du capteur LoRa

Ce programme permet d'envoyer une information via le LoRa:

1. Déclare une variable de type pointeur de caractère nommée cmdData et lui attribue la valeur "AT".
2. Déclare une variable de tableau de caractères nommée cmdDataTmp avec une taille de 40 caractères.
3. Utilise la fonction sprintf pour formater la chaîne de caractères "AT" avec un retour à la ligne ("\r\n") et la stocke dans la variable cmdDataTmp.
4. Appelle la fonction HAL_UART_Transmit avec les paramètres suivants :
 - L'adresse de la structure huart1
 - Le tableau charTransmit contient les données à transmettre.
 - La valeur 1, indiquant le nombre de caractères à transmettre.
 - Un délai d'attente de 1000 millisecondes (1 seconde).

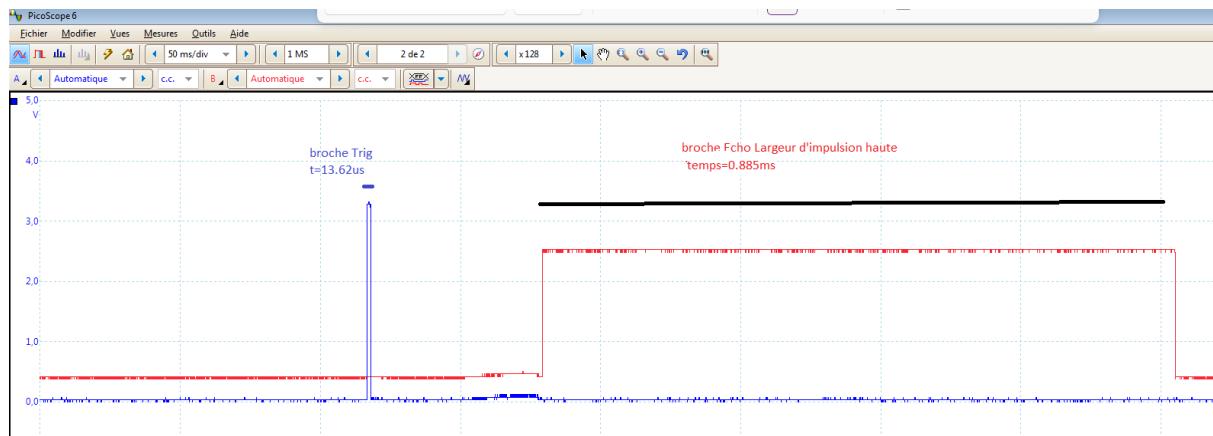
```
uint8_t charTransmit[1];
char *cmdData= "AT";
char cmdDataTmp[40];
sprintf(cmdDataTmp, "%s\r\n", cmdData);

HAL_UART_Transmit(&huart1, charTransmit, 1, 1000);
```

G. visualisation picoscope

On constate que le picoscope correspond bien à la mesure que nous avons effectué

on une largeur d'impulsion de 13 us trig en bleu qui correspond à la doc technique du constructeur et on a la largeur d'impulsion qui correspond à l'Écho qui est égale à 0.885 ms ce qui correspond à une distance de $d=340 \cdot 0.000885 = 30 \text{ cm}$



H. Validation Expérimentale

On constate qu'on a bien la valeur sur l'écran LCD qui est = 30 cm donc notre schéma électrique et notre programme marche comme il faut.



III. Conclusion

En conclusion, ce projet a été réalisé avec succès en utilisant le microcontrôleur STM32L476rg, le capteur de distance HC-SR04 et l'écran LCD. J'ai configuré les ports du microcontrôleur pour établir la communication I2C avec l'écran LCD et attribué une broche spécifique pour la communication One Wire avec le capteur HC-SR04. De plus, j'ai mis en place une fonction de temporisation en microsecondes pour assurer le bon fonctionnement du capteur. En suivant les spécifications détaillées dans la documentation technique (datasheet) du HC-SR04, j'ai pu initialiser le capteur et obtenir les données de distance requises. J'ai effectué les vérifications nécessaires conformément au protocole spécifié, et j'ai combiné les octets reçus pour obtenir les valeurs finales de distance.